

## CLAIMS

We claim:

1. An abstraction layer for interfacing a computer to a telephony  
5 radio, comprising:

a set of APIs for abstracting out multiple radio technologies without  
knowledge of the underlying radio technology, wherein the set of APIs  
correspond to call control functions, wherein when one of the set of APIs is  
called, the abstraction layer determines at least one standard telephony radio  
10 command corresponding to the called API and sends the telephony radio  
command to the telephony radio.

2. The abstraction layer of claim 1 wherein the telephony radio is  
one of a plurality of telephony radios which operates based on the standard  
15 telephony radio commands.

3. The abstraction layer of claim 1 wherein the set of APIs further  
correspond to short messaging system functions.

20 4. The abstraction layer of claim 3 wherein the set of APIs further  
correspond to network service functions.

5. The abstraction layer of claim 4 wherein the set of APIs further  
correspond to data connection functions.

6. The abstraction layer of claim 5 wherein the set of APIs further correspond to interface functions.

7. A radio interface layer of a telephone for facilitating  
5 communications between an application program module and a radio, comprising:

a proxy layer for communicating with the application program and a driver layer, wherein the application program calls an API to perform a particular function and wherein the proxy layer transforms the API to an  
10 IOCTL and sends the IOCTL to the driver layer; and

wherein the driver layer communicates with the proxy layer and the radio, the driver layer receiving an IOCTL and transforming the IOCTL into a command understood by the radio to perform the particular function.

8. The radio interface layer of claim 7 wherein the driver layer  
15 further receives communications from the radio indicating that the particular function has been performed and wherein the driver layer sends a success code to the proxy layer indicating that the particular function has been performed.

20

9. A method for processing commands in a telephone comprising a proxy layer, a driver layer, an application and a radio, the method comprising the steps of:

5 causing the application to call a RIL API in the proxy layer, wherein the RIL API is associated with an action to be performed by the radio;  
causing the proxy layer to translate the RIL API into input/output control (IOCTL) codes;  
sending the IOCTL codes to the driver layer;  
10 translating the IOCTL codes to a command corresponding to the action, wherein the command will be understood by the radio; and  
sending the command to the radio.

10. The method of claim 9 wherein the command is an AT  
15 command.

11. The method of claim 9 wherein the command is one of a private API set defined by the radio manufacturer.

20 12. The method of claim 9 further comprising the step of generating in the RIL driver layer a unique ID associated with the RIL API.

13. The method of claim 12 further comprising the step of waiting for a response from the radio, and when received, calling back the calling application with the response and the unique ID returned from the call.

5 14. The method of claim 13 wherein the RIL driver matches the response from the radio with the unique ID and the RIL driver sends the response to the calling process via a callback function.

15. A method of communicating between a client process and a server process in a distributed processing system of a telephone, comprising the steps of:

- 5           (a) issuing, by the client process, a call to the server process;
- (b) receiving, by the server process, the call and attempting to locate a radio of the telephone in response to the call; and
- (c) determining, by the server process, that the radio is not present and sending the error code **RIL\_E\_RADIONOTPRESENT** to the
- 10   client process.

16. A method of communicating between a client process and a server process in a distributed processing system, comprising the steps of:

- (a) issuing, by the client process, a call to the server process;
- 5 (b) receiving, by the server process, the call, locating a radio of the system and executing the call; and
- (c) determining, by the server process, in executing the call, that the radio has been removed and sending the error code **RIL\_E\_RADIOREMOVED** to the client process.

10

17. A method of communicating between a module and a radio comprising:

(a) generating a RIL API call at one of a plurality of modules to perform a specific action;

5 (b) sending the RIL API call to a radio driver;

(c) at the radio driver, converting the RIL API call to a command understood by the radio;

(d) transmitting the command to the radio; and

(e) performing the specific action at the radio.

10

18. The method of claim 17 further comprising:

(f) in response to successfully performing the specific action, sending a success code to the one of the plurality of modules that generated the RIL API.

15

19. The method of claim 18 wherein the RIL API, command and success code are associated with an identifier linking them together and linking them to the one of the plurality of modules that generated the RIL API call and wherein the radio driver receives the success code, and, using the identifier, matches the success code with the one of the plurality of modules that generated the RIL API call and sends the success code to the one of the plurality of modules that generated the RIL API call.

20

20. The method of claim 19 further comprising the step of:  
(g) generating a notification at the radio in response to detecting data that needs to be reported to one of the plurality of modules;  
5 (h) sending the notification to the radio driver.

21. The method of claim 20 further comprising the step of:  
(i) sending the notification from the radio driver to one of the plurality of modules.

10 22. The method of claim 20 further comprising the step of:  
(i) sending the notification from the radio driver to each of the plurality of modules.

15 23. The method of claim 20 wherein the data that needs to be reported comprises an incoming phone call to the radio.

24. The method of claim 20 wherein the data that needs to be reported comprises a signal strength change in the radio.

20 25. The method of claim 20 wherein the one of a plurality of modules is a TSP.



26. The method of claim 20 wherein the one of a plurality of modules is a SIM manager.

27. The method of claim 20 wherein the one of a plurality of modules is an emergency application for generating emergency calls.

28. The method of claim 20 wherein the one of a plurality of modules is a WAP layer.

29. The method of claim 20 wherein the one of a plurality of modules is a TAPI interface.

30. The method of claim 20 wherein the one of a plurality of modules is an ExTAPI interface.

31. The method of claim 20 wherein the one of a plurality of modules is connected to an application program module and receives instructions from the application program module to generate the RIL API call.

32. The method of claim 31 wherein the instructions provided by the application program module comprise instructions defined by the one of a plurality of modules and wherein the instructions are converted to the RIL API calls by the one of a plurality of modules.